Funambol Data Synchronization Server

Architectural Overview

Funambol Technical Paper TP100
March 2006

Important Information

Published by Funambol, Inc., 643 Bair Island Road, Suite 305, Redwood City, CA 94063

Introduction

The purpose of this overview is to provide a technical foundation for architects and developers to understand how the Funambol synchronization server is built. Because it is intended for the software developer, we start at the core of the engine and work out towards the edge and the user features.

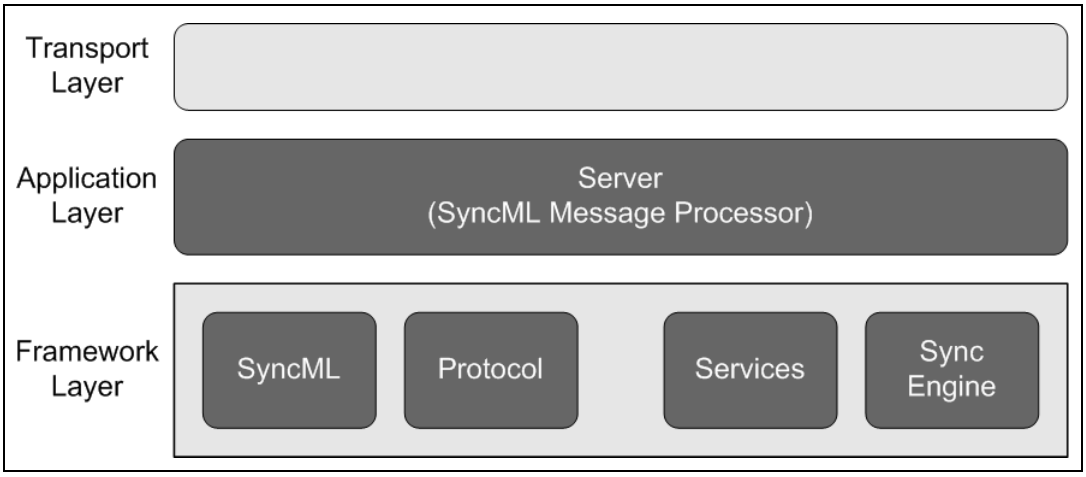
This overview should be supplemented with several other documents in the Funambol library, in particular:

- DS Server Quick Start Guide
- DS Server Developer's Guide
- DM Server Developer's Guide

These and other documents, plus the source code itself, can be downloaded from `www.funambol.com/opensource`.

Core Architecture

The Funambol core architecture is implemented in layers, as shown below:

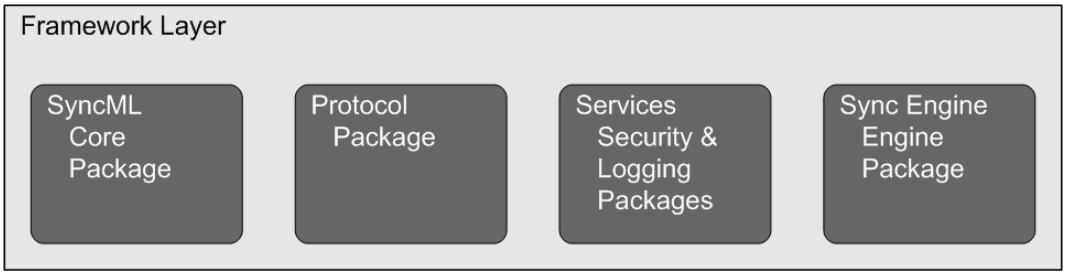


The framework layer implements and provides protocols, horizontal services, and the synchronization engine, on top of which the transport and application layers are developed. The application layer consists of the server that accepts and processes SyncML messages.

The server relies on the transport layer to receive messages delivered using different protocols, such as HTTP, SMTP, OBEX, and so on. Currently, the server is implemented as an Enterprise Java Bean (EJB) service deployable in a J2EE-compliant application server.

The Framework Layer

The framework layer consists of several packages, the most important of which are shown below:



Core Package – the foundation classes used to represent a message. This module provides for the translation of an XML stream into an object tree; conversely, an object representing a message can be converted into the corresponding XML representation. These classes check that a message is a valid SyncML message; however, this validity check guarantees only that the XML structure can represent a message, regardless of the context in which the message is processed. The scope of this check is to verify that the SyncML representation rules are observed.

Protocol Package – a SyncML communication is a sequence of correlated messages that must follow additional rules, as specified by the SyncML protocol. This module handles the processing of such messages to ensure compliance with the protocol.

Security and Logging Packages – this module implements logging and security services. For the security aspects, Funambol adheres to the Java Authentication and Authorization Service (JAAS) delivered with JDK 1.4. You can develop and use your own authentication and authorization policy independent of that implemented by Funambol.

Engine Package – provides the logic for the synchronization server. The default implementation provides a set of rules, including the following:

- Identify the source and destination of the data to be synchronized.
- Identify the data that needs to be updated, added, or deleted.
- Determine how updates are to be applied.
- Detect and resolve conflicts.

Funambol also allows developers to deploy their own implementation of the synchronization engine to extend the standard functionality to meet their own requirements. Developers can even completely replace the default implementation with a custom engine.

The engine package also provides a basic interface for the synchronization engine. It defines solely the interface, this maintains the pluggable architecture for the engine. The process of receiving and interpreting a synchronization message, and the process of updating the data source and producing the modifications for the client, are two distinct processes. These processes can be applied independently from one another. For example, from the synchronization point of view, it does not matter if a synchronization request came from a SyncML message or a simple HTTP request. Likewise, from the protocol point of view, it does

not matter which conflict resolution strategy the synchronization engine adopts. With this pluggable architecture, the business logic of the protocol and of the actual synchronization can be developed and extended separately, without touching the server or the other modules.

Other Packages – In addition to the above packages, a client and server package provide common classes for the development of client and server applications.

The Application Layer

Packages are provided for clients and the synchronization server. The server is implemented as an Enterprise Java Bean (EJB) that can be deployed in a J2EE-compliant application server. This design provides the following benefits:

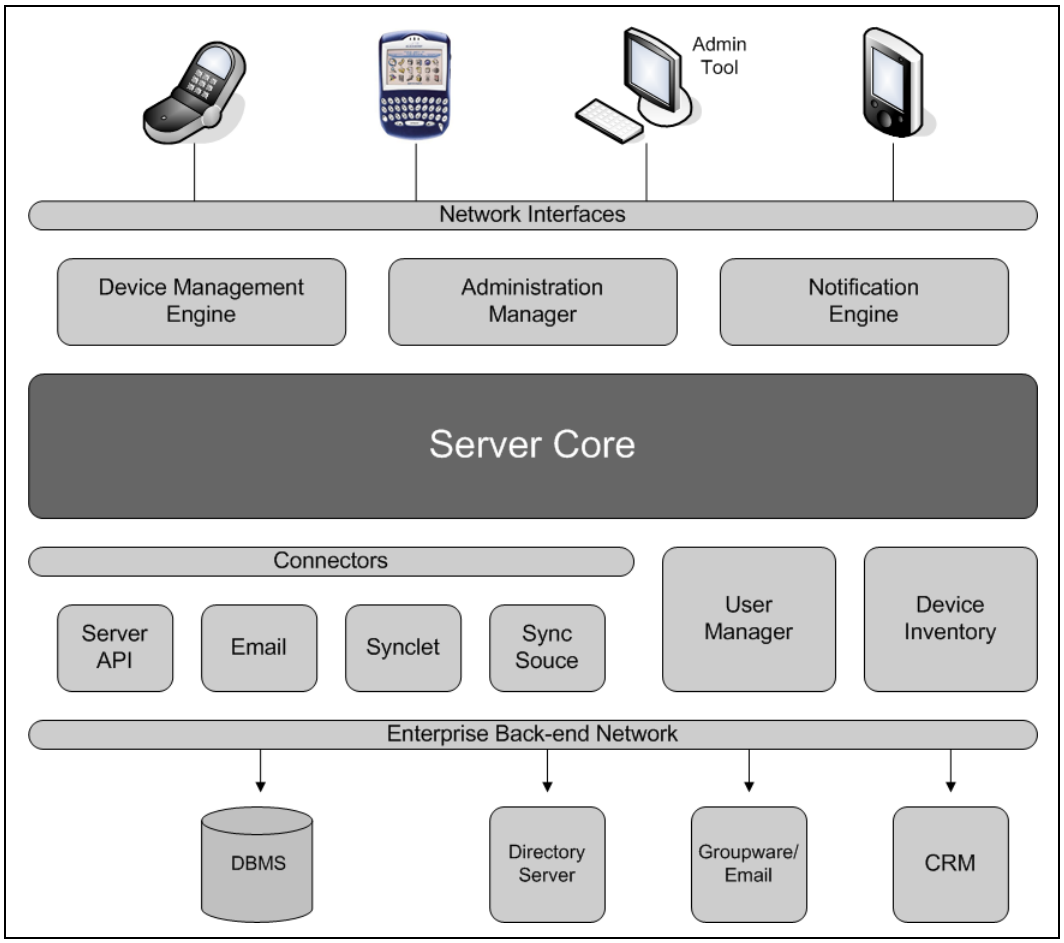
- It uses a widely-accepted standard.
- The transport protocol is decoupled from the synchronization logic.
- Application servers provide many out-of-the-box facilities and services that otherwise would need to developed, including connection management, thread management, security, scalability, availability, and reliability.
- Use of the existing application server infrastructure simplifies management and deployment.

The Transport Layer

Currently, Funambol supports the HTTP protocol for message transport. Other transport protocols will be added in future releases.

System Architecture

Funambol provides the following components around the server core:



- **Device Management Engine** – a component that handles SyncML DM requests.
- **Administration Manager** – a component that manages users, devices, principals, and sync sources.
- **Notification Engine** – a component that handles server-alerted synchronization.
- **Connectors** – server extensions that integrate the server core with external sources of data.
- **User Manager** – a component that manages users.
- **Device Inventory** – a registry of devices associated with the system.

Conclusion

This overview was intended to describe the elements and basic functions of the major subsystems of the Funambol synchronization server. The server is typically deployed inside a J2EE application server, and interacts with connectors running on other servers (to get access to corporate data sources) as well as the Administration Tool that runs on a PC.

For a full list of features and platform support information, download a copy of the product data sheet from `www.funambol.com/product/documentation`.